

NON-NATIVE ENGLISH ACCENT DETECTION

A CAPSTONE PROJECT REPORT
BY

APAR GARG (E17CSE112)



SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

in

*Partial fulfillment of the requirements
for the degree of*

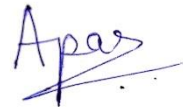
BACHELOR OF TECHNOLOGY

NOVEMBER 2020

CERTIFICATE

This is to certify that the capstone project report entitled “**Non-Native English Accent Detection** ” is being submitted by **Mr. Apar Garg** (Enroll. No. E17CSE112) to the Department of Computer Science Engineering, Bennett University, Greater Noida, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology. It is an original research work carried out by him/them as the 7th Semester 12 credit course from June 2020 to November 2020.

The report has fulfilled all the requirements as per the regulations of this institute and has reached the standard needed for submission. The results embodied in this capstone project report has not been submitted to any other university or institute for the award of any other degree or diploma, degree elsewhere.



Apar Garg

(Enroll. No. E17CSE112)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Indrajeet Gupta (Capstone Project Instructor)

Assistant Professor

Computer Science Engineering Department

Bennett University, Greater Noida, INDIA

DECLARATION

I hereby declare that the work which is being presented in the report entitled “**Non-Native English Accent Detection**”, in partial fulfillment of the requirements for the Bachelor of Technology in Computer Science and Engineering is an authentic record of my work carried out during the period from August 2020 to November 2020 at Department of Computer Science and Engineering, Bennett University Greater Noida.

The matters and the results presented in this report have not been submitted by me for the award of any other degree elsewhere.



Apar Garg

(Enroll. No. E17CSE112)

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to my mentor, **Dr. Tanmay Bhowmik, Assistant Professor, Computer Science Engineering Department, Bennett University** for guiding, supporting, and helping me in every possible way. I was extremely fortunate to have him as my mentor as he provided insightful solutions to problems faced by me thus contributing immensely towards the completion of this capstone project. I would also like to express my deepest gratitude to VC, DEAN, HOD, faculty members, and friends who helped me in the successful completion of this capstone project.



Apar Garg

(Enroll. No. E17CSE112)

LIST OF ABBREVIATIONS

Abbreviation	Explanation of the Abbreviation
ML	Machine Learning
DL	Deep Learning
MFCC	Mel-frequency Cepstral Coefficient
KNN	K-Nearest Neighbors
CNN	Convolutional Neural Network
AutoML	Automated Machine Learning
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
ZCR	Zero-crossing rate

ABSTRACT

Nowadays, home assistants like Apple Siri and Google Home have become an integral part of our lives as they make mundane tasks such as setting up reminders and checking emails easy. Non-native English speakers frequently face problems with these automated assistants because of accented speech. I have proposed a method to identify speech samples with a non-native English accent, which can then be logically acted upon accordingly to reduce these barriers resulting from accent distinctness. This speech command could then be easily understood by home assistants. For accent classification, I have extracted various spectral features from speech samples. Then, I have applied various Machine Learning and Deep Learning models and achieved an accuracy of 86.67%.

1. INTRODUCTION

These days, systems like Apple Siri and Google Home are rapidly making speech classification in the household a more feasible and affordable technology. Home assistants can handle numerous tasks such as checking emails, scheduling a wake-up call, and buying products, etc. Unfortunately, when trying to converse with automated assistants, non-native English speakers frequently face problems as accented speech often involves phones or sounds that are not typical in normal pronunciation [3]. Measures are therefore needed to be able to correctly identify a speaker's accent or locale, which can then be logically acted upon accordingly [1].

Accent classification is the method of using a person's different audio features to predict and classify his or her accent [2]. In areas like crime investigation, the Accent classification system, which classifies the origin or ethnicity of a speaker, is significant. Recognizing accents from short audio samples is critical in real-world applications [3].

Because of the various features that set accents apart, accent classification is a complicated problem. Accents vary according to voice quality, prosody, and pronunciation of phonemes. Since these particular features are hard to extract, current work utilizes alternative features, for example, spectral features. The MFCC and Spectral Centroid etc., which are derived from raw audio recordings, provide such features. MFCC is among the most widely used spectral feature representation in accent classification systems. Previous research indicates that MFCC used in conjunction with other spectral features may improve the accuracy of accent classification systems [4].

The objective of this study is to find the features for accent classification that will give the highest classification accuracy. I attempted to classify English accent as American or non-American using 6 unique spectral features namely MFCC Mean, Spectral Centroid Mean, ZCR Mean, Chroma Frequencies Mean, Spectral Bandwidth Mean, and Spectral Roll-off Mean. I used various Machine Learning models like SVM, Decision Tree, Random Forest, and Logistic Regression, etc. on these acoustic features of speech and compared the results obtained from them.

As it is the most commonly spoken language in the world, I have confined the work to the English language. I have further confined the work by taking the American accent in native English accents. Hence, any other accent beside the American English accent was taken to be a non-native accent. With sufficient relevant data available, using my models as a basis, similar solutions can also be given for other national and regional languages.

There are two main contributions to this study:

1. Comparison of various spectral feature sets on model accuracy.
2. Comparison of manual ML with AutoML.

2. SYSTEM SPECIFICATIONS

Item	Value
Processor	Intell Core I i5-8250U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)
Total Physical Memory	7.89 GB
Total Virtual Memory	31.9 GB
System type	x64-based PC
OS Name	Microsoft Windows 10 Home Single Language
OS Version	10.0.18363 Build 18363

Programming Language Used – Python

Platform – Jupyter Notebook, Sublime Text Editor

3. DATASET DESCRIPTION

The dataset used is The Speech Accent Archive [10]. This dataset consists of 2140 samples of speech, each from a unique speaker reading the same passage. Speakers have 214 distinct native languages and come from 177 countries. The speakers are reading the following lines in English-

"Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station".

This paragraph contains complicated consonant clusters such as P-S and S-N and nearly all vowels and consonants, making it the best choice for evaluating the accent of the speaker [6]. The audio samples have minimal background noise (desirable) and pause between words (undesirable) [5].

The demographic information of the people is also present viz., age, age of English onset, birthplace, native language, sex, and country. Table 1 shows the Data Type of each column in the dataset.

Table 1: Data Type of each demographic column in the dataset.

Data Type	Columns
Discrete (Numerical)	age, age onset
Nominal (Categorical)	birthplace, sex, country

In this study, I aimed to classify accent as American English or non-American English. Moving forward, I have used the terminologies as mentioned in Table 2.

Table 2: Standard Terms used in the paper.

Actual Term	Term in Paper
American English accent	native accent
non-American English accent	non-native accent

4. PROPOSED METHODOLOGY

4.1. Accent Classification

Figure 1 gives an overview of my proposed methodology. The audio dataset was first augmented to handle class imbalance. Then audio samples were preprocessed into uniform length (20 seconds) audio files and segmented into 4 sec samples. Then I extracted 6 unique spectral features from audio samples namely MFCC Mean, Spectral Centroid Mean, ZCR Mean, Chroma Frequencies Mean, Spectral Bandwidth Mean, and Spectral Roll-off Mean. Later Correlation-Based Feature Selection was performed, and the remaining features were used for training the models.

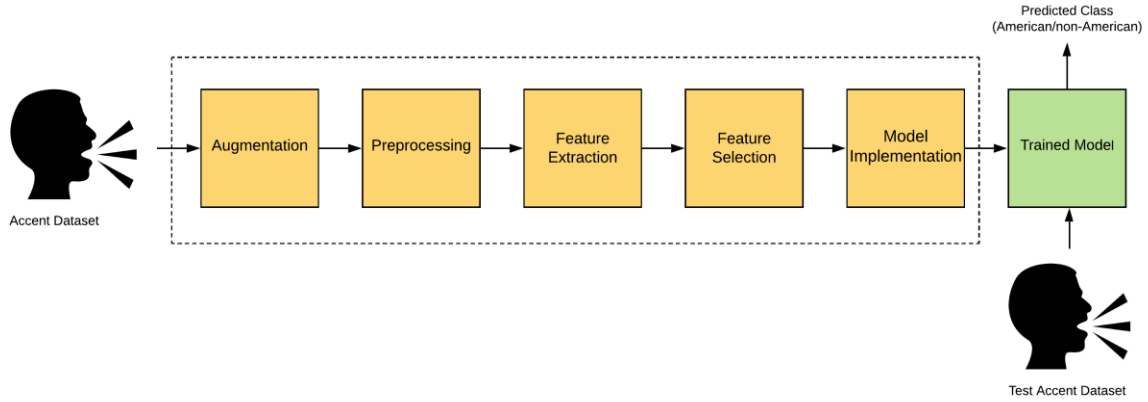


Figure 1: Overview of methodology.

4.1.1. Target variable

A new target variable was created in which all rows in the dataset having country as 'usa' and native language as 'english' were labeled as 1(native accent) and the rest were labeled as 0(non-native accent). All the ML and DL models used this variable as a dependent feature for training. Table 3 shows the distribution of native and non-native accents in the dataset.

Table 3: Number of audio samples per accent class.

Accent	Number of samples
Native accent	373
Non-native accent	1767
Total	2140

Since none of the demographic features seemed relevant to be used directly as independent features for accent classification, they were dropped from the dataset.

4.1.2. Data Augmentation

From Table 3, I observed that the dataset had unbalanced classes. To solve this problem, I used data augmentation. Data augmentation is a technique used to generate synthetic data, i.e. creating new samples in the given samples by adjusting small factors [11].

A sound wave has the following characteristics: Pitch, Loudness, Quality. I need to alter my samples around these characteristics in such a way that they only differ by a small factor from the original sample. I applied noise injection, shifting time, changing pitch, and speed with the help of NumPy [12] and librosa [13] to handle class imbalance. Table 4 show the distribution of native and non-native accent in the dataset after data augmentation. Figure 2 present a comparison of the number of instances in each class of dataset before and after augmentation, respectively.

Table 4: Number of samples generated for each class using augmentation and the number of samples selected for further processing.

Accent	Number of samples generated	Number of samples selected
Native accent	1865	1865
Non-native accent	8835	2000
Total	10700	3865

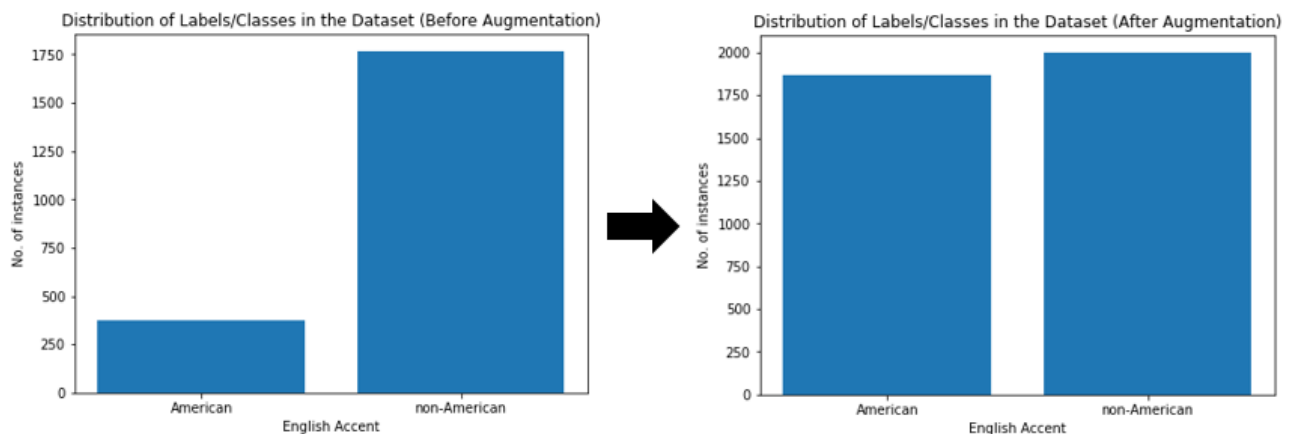


Figure 2: Number of instances/samples in each target class before and after augmentation.

4.1.3. Preprocessing

There were 3865 audio samples of varying lengths in the resulting audio dataset. But I needed all samples to be of the same length to use the data for model implementation. In this proposed work I padded the shorter samples and trimmed the longer samples to 20 sec, thereby making all samples of equal length. The remaining speech samples were then trimmed into multiple segments with an equal length of 4 sec. Thus every 20 sec speech sample was segmented into 5 parts.

Therefore, as each 20 sec (let's call it *track_duration*) audio file was sampled at 22050 Hz (let's call it *sample_rate*) to get the one-dimensional NumPy array (let's call it *y*), the pseudo-code to divide the file into segments is as shown in Figure 3. Table 5 presents the number of instances in each target class after segmentation.

```
samples_per_track = sample_rate * track_duration
samples_per_segment = samples_per_track / num_segments
for segment_number = 0 to 5:
    start = samples_per_segment * segment_number
    finish = start + samples_per_segment
    ysegment = y[start:finish]
```

Figure 3: pseudo-code for segmentation of a given audio sample.

Table 5: Number of samples in each target class after segmentation.

Accent	No. of samples
Native accent	9325
Non-native accent	10000
Total	19325

4.1.4. Feature Extraction

For accent classification, one can either use the spectrograms directly or extract the acoustic features from the speech signal and use the classification models on them. In my study, I chose to extract the features for classification [14].

6 unique spectral features were extracted from each of the 19325 speech samples for this study. I extracted the means of 13 Mel-frequency cepstral coefficients (MFCC). Since the first 13 MFCC coefficients reflect the spectral envelope and envelopes are sufficient to distinguish different phonemes, they are usually taken as features. The discarded higher dimensions give the spectral details [15]. I also extracted the means of some other spectral features namely Spectral Centroid, Zero Crossing Rate, Chroma Frequencies (using fixed-window STFT analysis), Spectral Bandwidth, and Spectral Roll-off. Therefore, I extracted a total of 18 spectral features.

Figure 4 visualizes the audio signal of a native accent speaker reciting a part of a paragraph from the dataset i.e. "Please call Stella". For visualization purposes, I have chosen the default parameters provided by librosa.

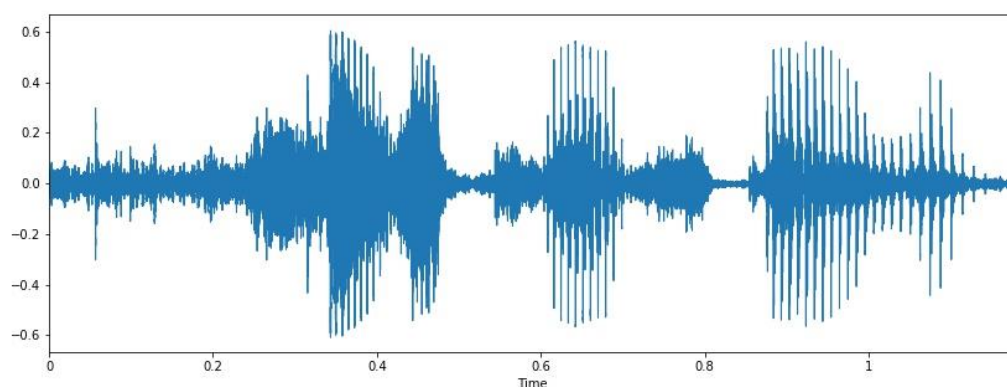


Figure 4: Original audio signal (Amplitude vs Time) of a user.

4.1.4.1. *MFCC*

MFCC analysis is the reduction of an input speech signal into critical speech component features by utilizing techniques like Mel-frequency analysis and Cepstral analysis. Cepstral analysis in conjunction with Mel-frequency analysis gives 12 or 13 MFCC features related to speech. Optionally, the Delta and Delta-Delta MFCC features may be applied to the feature set, potentially increasing the number of features, up to 39 features, but delivering stronger ASR performance [7].

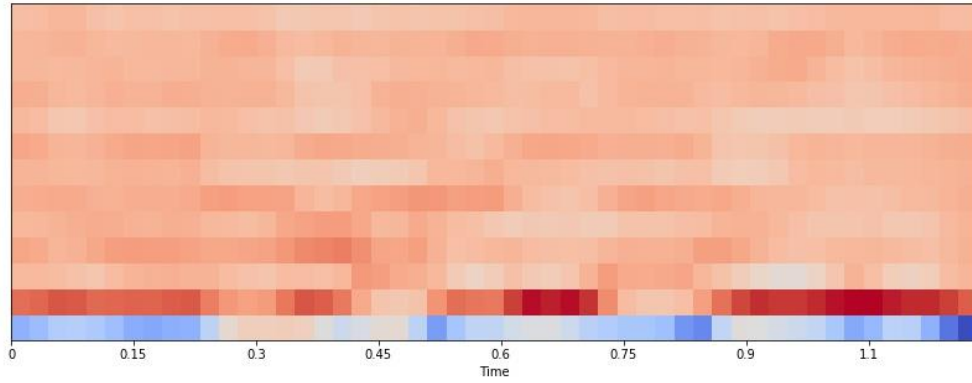


Figure 5: 13 MFCC Representation (Frequency vs Time) of Figure 4.

4.1.4.2. *ZCR*

ZCR tells the number of times the amplitude of a speech signal passes from the zero value in a given time frame/interval. Therefore, ZCR equals 4 in Figure 6. If there are many zero crossings, it means that there is no dominant low-frequency oscillation.

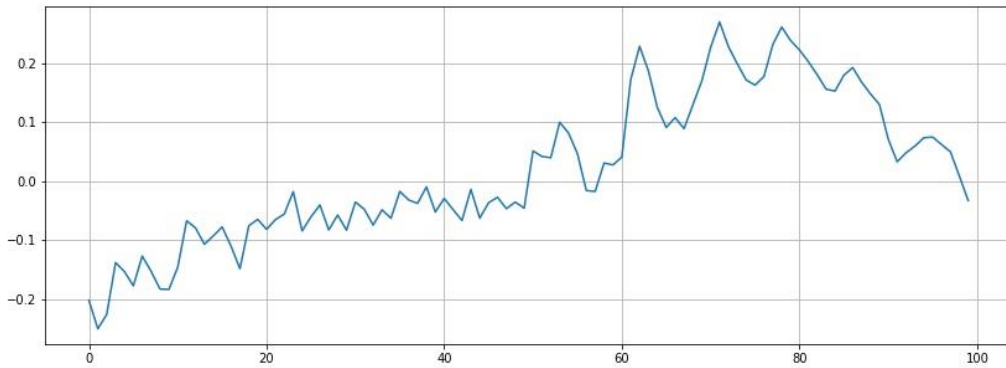


Figure 6: ZCR Representation of Figure 4.

4.1.4.3. *Spectral Centroid*

Spectral Centroid is the spectrum's center of mass. The spectrum gives details on the amplitude (mass) of the signal transmitted between frequencies [7]. It gives the weighted average/mean of the sound frequencies using the formula:

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)}$$

where $S(k)$ is the spectral magnitude and $f(k)$ is the frequency at frequency bin k .

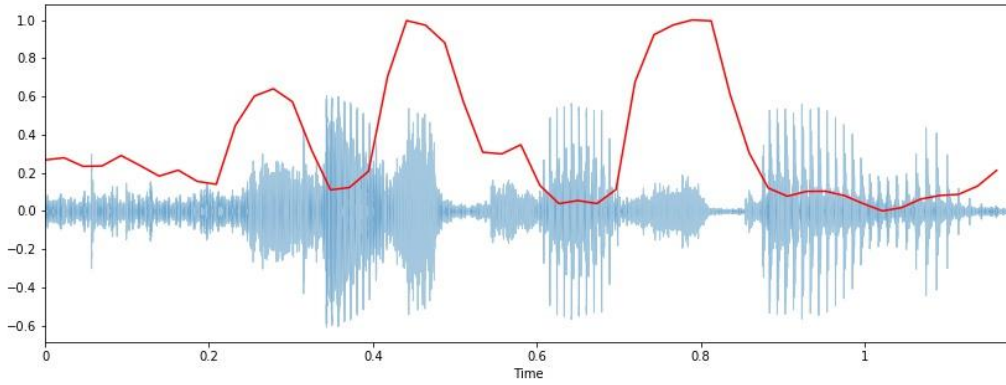


Figure 7: Spectral Centroid Representation of Figure 4.

4.1.4.4. *Spectral Roll-off*

Spectral Roll-off is a measure of the shape of the signal. It is identical to Spectral Centroid. The weighted average/mean of the frequency amplitude is determined by Spectral Centroid, but Spectral Roll-off gives the frequency below which a given percentage of the total spectral energy lies, e.g. 85 percent [7].

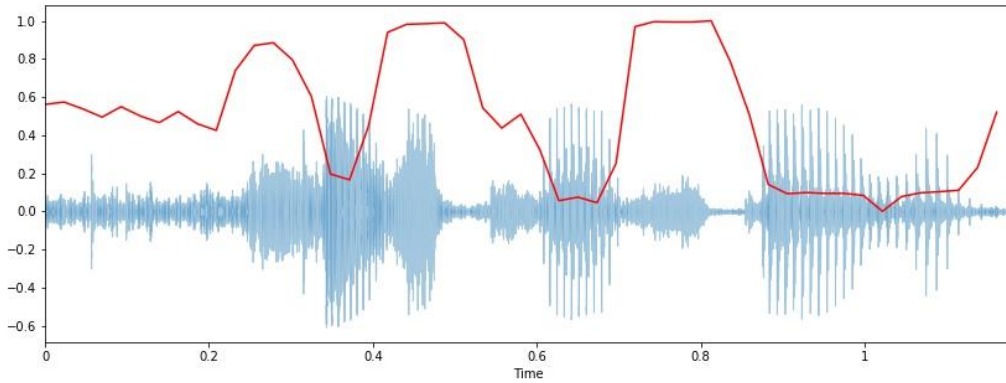


Figure 8: Spectral Roll-off Representation of Figure 4.

4.1.4.5. *Spectral Bandwidth*

The librosa function for spectral bandwidth calculates the order- p spectral bandwidth using the following formula:

$$\left(\sum_k S(k)(f(k) - f_c)^p \right)^{\frac{1}{p}}$$

where $S(k)$ is the spectral magnitude and $f(k)$ is the frequency at frequency bin k . f_c is the spectral centroid. When $p=2$, this is like a weighted standard deviation.

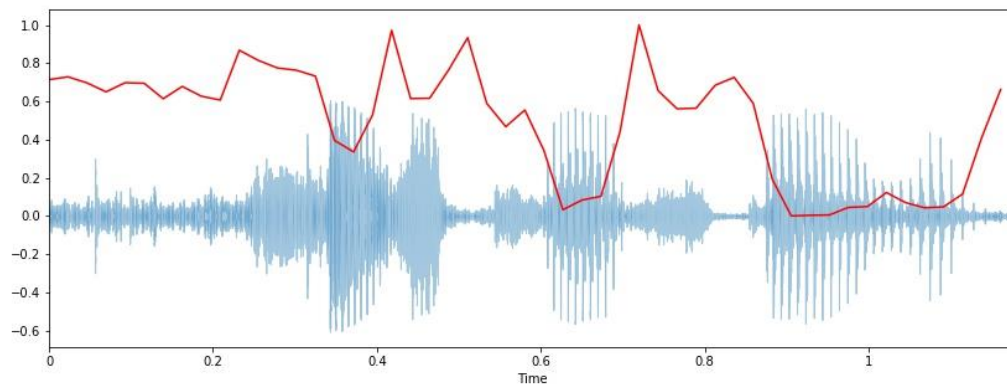


Figure 9: Spectral Bandwidth Representation of Figure 4.

4.1.4.6. *Chroma features*

Chroma features are a fascinating and effective music audio representation in which the entire spectrum is projected across 12 bins depicting the musical octave's 12 different semitones (or chroma) [7].

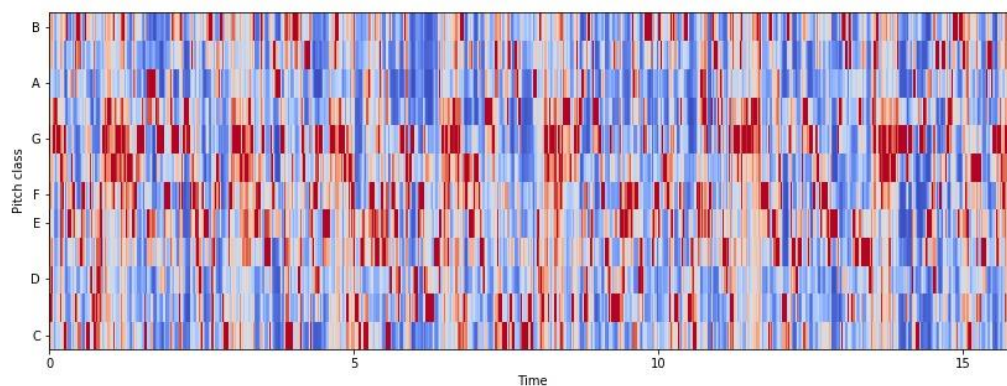


Figure 10: Chromagram Representation of Figure 4.

For this study, frame and hop lengths are set to 400 and 100 samples, respectively for all spectral features. At the sampling rate of 16000 Hz, this corresponds to overlapping frames of approximately 25ms spaced by 6.25ms. The rest of the parameters were set to default (as provided by librosa).

4.1.5. Model Implementation

Figure 11 shows the pipeline I have followed for model implementation. I have implemented the ML models manually as well as with AutoML techniques.

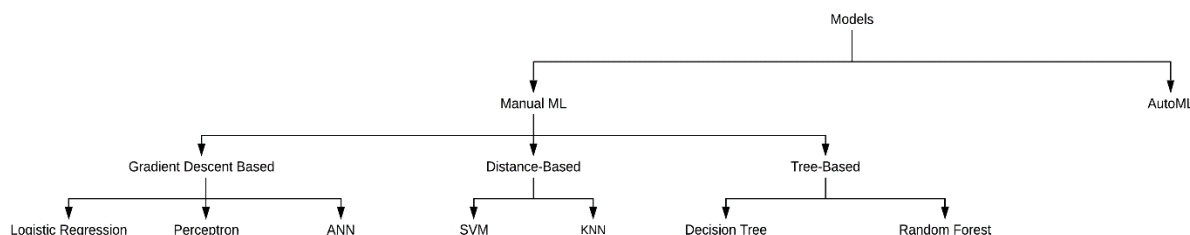


Figure 11: Classification of ML models implemented.

4.1.5.1. *Manual ML*

Feature scaling is one of the most important data pre-processing steps in machine learning. It is used to normalize the range of independent variables or features of data. I trained the models on unscaled, normalized, and standardized data and compared the performance for best results.

As evident in Figure 11, I used both traditional ML and DL algorithms to classify accent as native or non-native. I used the functions in the scikit-learn [8] package. Table 6 shows the functions in scikit-learn corresponding to each model used for classification.

Table 6: scikit-learn functions for every model.

Model Class	Model	Scikit Learn function
Distance-Based	KNN	KNeighborsClassifier
	SVM Classifier	SVC
Gradient Descent Based	Logistic Regression	LogisticRegression
	Perceptron	Perceptron
	ANN	MLPClassifier
Tree-Based	Decision Tree	DecisionTreeClassifier
	Random Forest	RandomForestClassifier

GridSearchCV method was used to fine-tune the hyperparameters in all the models. The models were trained on all possible combinations of:

- Features –
 1. 13 MFCC Mean,
 2. Spectral Centroid Mean+ ZCR Mean + Chroma Frequencies Mean + Spectral Bandwidth Mean + Spectral Roll-off Mean,
 3. All 18
- Scaling – Unscaled (Original), Normalization, Standardization

4.1.5.2. *AutoML*

AutoML refers to the process of automating an entire or a part of the ML pipeline. I used TPOT for AutoML. TPOT is an AutoML tool in python that utilizes genetic programming for optimizing machine learning pipelines. It explores a number of possible pipelines to find the right one for the data provided, it automates the most boring part of ML [9].

I trained the TPOTClassifier function in TPOT on the unscaled dataset for accent classification. All the parameters were set to default for training, except for generations and population_size. Generations and population_size were set to 7 and 70 respectively. Therefore, TPOT searched through 560 (population_size + generations x offspring_size) pipelines in total and selected the best pipeline which fitted the unscaled dataset.

5. RESULTS

5.1. Accent Classification

5.1.1. Test Dataset

I conducted a pilot evaluation task for the Accent Classification system that included a group of 39 random individuals amongst which 15 were native and 24 were non-native. I asked each of them to record and send multiple speech samples of a minimum of 20 sec length and having random transcriptions. A total of 345 samples were received. Then the pseudo algorithm as mentioned in Figure 3 was applied to segment the samples and after that spectral features were extracted from each 4 sec sample. The resultant dataset was reserved for testing purposes. Table 7 shows the distribution of native and non-native accents in the testing dataset.

Table 7: Number of samples in each target class of test dataset.

Accent	Number of samples
Native accent	785
Non-native accent	938
Total	1723

5.1.2. Manual ML

ANN gave the highest accuracy of 80% on Standardized dataset. Table 8 shows the Precision, Recall, and F1 score for each accent class in ANN. Table 9 shows the confusion matrix for the ANN. Table 10 presents the hyperparameters chosen by GridSearchCV for the ANN.

Table 8: Precision, Recall, and F1 score for each class of prediction.

Accent	Precision	Recall	F1-score
non-native	0.79	0.85	0.82
native	0.80	0.73	0.76

Table 9: Confusion matrix for ANN.

Predict \ Actual	non-native	native
non-native	794	144
native	212	573

Table 10: Hyperparameter values selected by GridSearchCV for ANN.

Hyperparameter	Value
hidden_layer_sizes	(20,20,20)
activation	relu
solver (optimizer)	adam
alpha (L2 regularization)	0.0001
learning_rate	adaptive

5.1.3. AutoML

Table 11 shows the testing accuracy of the TPOTClassifier when trained on different feature sets of the test dataset. Table 12 shows the Precision, Recall, and F1 score for each accent class when trained on all 18 features. Table 13 shows the confusion matrix for all same.

Table 11: Summary of the best results from the TPOTClassifier.

Features	Test Accuracy (%)
MFCC Mean	82.12
All 18 features	86.76

Table 12: Precision, Recall, and F1 score for each class of prediction.

Accent	Precision	Recall	F1-score
non-native	0.87	0.90	0.88
native	0.86	0.83	0.85

Table 13: Confusion matrix for ExtraTreesClassifier.

Predict \ Actual	non-native	native
non-native	2169	245
native	325	1569

As evident from Table 11, TPOT searched through 560 ML pipelines for each feature set and got the maximum testing accuracy of 86.76% on all 18 features. The best pipeline chosen by TPOT for 86.76% accuracy used MinMaxScaler preprocessor and ExtraTreesClassifier model on the dataset. Table 14 shows the hyperparameters of ExtraTreesClassifier.

Table 14: Hyperparameter values selected by TPOT for ExtraTreesClassifier.

Hyperparameter	Value
bootstrap	False
criterion	gini
min_samples_split	6
max_features	0.75
min_samples_leaf	2
n_estimators	100

6. CONCLUSION AND FUTURE WORK

In this study, we proposed a learning framework for classification of English accent as American or non-American by trying out a different set of acoustic features namely MFCC Mean, Spectral Centroid Mean, ZCR Mean, Chroma Frequencies Mean, Spectral Bandwidth Mean, and Spectral Roll-off Mean. From the results obtained it was noted that manual ML and AutoML models, when trained on MFCC Mean in conjunction with other features produced better results than when trained only on MFCC Mean. It was also noted that AutoML improved accuracy by 7.46%.

My future goal is to work on an open-source dataset with at least 10k audio samples. The main objective will be to perform a deeper analysis of the dataset. I will use other neural networks like RNN, and CRNN to perform classification.

REFERENCES

1. Accent Classification in Human Speech Biometrics for Native and Non-native English Speakers.
2. Foreign accent classification using deep neural nets Utkarsh Singha,*, Akshay Guptab, Dipjyoti Bisharadc and Wasim Arife
3. English Language Accent Classification and Conversion using Machine Learning Pratik Parikh1,*, Ketaki Velhal, Sanika Potdar, Aayushi Sikligar, Ruhina Karani
4. Y. Singh, A. Pillay and E. Jembere, "Features of Speech Audio for Accent Classification," 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2020, pp. 1-6, doi: 10.1109/icABCD49160.2020.9183893.
5. English Language Accent Classification and Conversion using Machine Learning Pratik Parikh1,*, Ketaki Velhal2, Sanika Potdar3, Aayushi Sikligar4, Ruhina Karani5
6. Speech Recognition Learning Framework for Non-Native English Accent
7. Features of Speech Audio for Accent Recognition
8. Scikit Learn Package. <http://scikit-learn.org/stable/>
9. Olson R.S., Urbanowicz R.J., Andrews P.C., Lavender N.A., Kidd L.C., Moore J.H. (2016) Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. In: Squillero G., Burelli P. (eds) Applications of Evolutionary Computation. EvoApplications 2016. Lecture Notes in Computer Science, vol 9597. Springer, Cham. https://doi.org/10.1007/978-3-319-31204-0_9
10. Weinberger, S. (2013). Speech accent archive. George Mason University.
11. Ko, Tom, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. "Audio augmentation for speech recognition." In Sixteenth Annual Conference of the International Speech Communication Association. 2015.
12. Oliphant, Travis E. A guide to NumPy. Vol. 1. USA: Trelgol Publishing, 2006.
13. McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python." In Proceedings of the 14th python in science conference, vol. 8, pp. 18-25. 2015.
14. <https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8>
15. <https://towardsdatascience.com/how-i-understood-what-features-to-consider-while-training-audio-files-eedfb6e9002b>